On Simultaneous Selection of Prototypes and Features in Large Data

T. Ravindra Babu[1], M. Narasimha Murty[1], and V.K. Agrawal[2]

[1] Department of Computer Science and Automation,
Indian Institute of Science,
Bangalore, India
[2] ISRO Satellite Centre, Bangalore, India

Abstract. In dealing with high-dimensional, large data, for the sake of abstract generation one resorts to either dimensionality reduction or cluster the patterns and deal with cluster representatives or both. The current paper examines whether there exists an equivalence in terms of generalization error. Four different approaches are followed and results of exercises are provided in driving home the issues involved.

Keywords: Data Mining, prototype selection, frequent itemsets, clustering,feature selection.

1 Introduction

In a broad sense, any method that incorporates information from training samples in the design of a classifier employs learning [1]. Classification of Large Data is a challenging task, especially in the context of Data Mining. Occam's famous razor states that "Entities should not be multiplied beyond necessity". One may restate this as, "Given two models with the same generalization error, the simpler one should be preferred because simplicity is desirable in itself" [2].

With set of training patterns, each characterized by a large number of features the phenomena of "curse of dimensionality"[1] dominates. For a 'd' binary-featured pattern, the total number of possible distinct patterns is 2^d. This makes a classification algorithm unwieldy with large data, which is the focus of the current work. In view of this, one resorts to dimensionality reduction and less complex models, even though it amounts to discarding some information. At the same time, one would look to avoid overfitting and achieve smoother discrimination region. Alternatively, one can also find prototypes from the data by resorting to clustering. The approaches followed in the current work are novel and different from earlier work on the topic [8].

In the current work, we classify large Handwritten Digit data by four approaches combining dimensionality reduction and prototype selection. The compactness achieved by dimensionality reduction is indicated by means of number of combinations of distinct subsequences [5]. The concepts of Frequent items [3]

S.K. Pal et al. (Eds.): PReMI 2005, LNCS 3776, pp. 595–600, 2005.

and Leader cluster algorithms [4] are made use in the work. The k-NNC is the learner[1].

2 Training, Validation and Test Data

Large handwritten digit data of 10 classes is considered for the study. The data consists of 10003 labeled patterns, with 6670 training and 3333 test patterns. Each pattern consists of 192 binary features. The number of patterns per class is almost uniform. The above training data is further subdivided, for the current study, into training (6000) and validation (670) data. Figure 1 contains a sample set of handwritten data, which is under study.

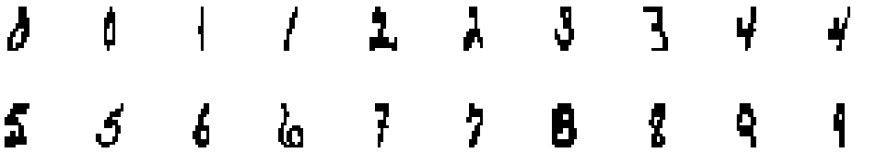

Fig. 1. A sample set of patterns of Handwritten data

3 Basic Terminology

Consider training data containing 'm' patterns, each having 'p' features. Let 'ϵ' and 'η' be minimum support [6, 3] for considering any feature for the study and distance threshold respectively. Definitions of parameters which are used in the rest of paper are provided below.

Definition 1 ***Sequence[7].*** *A sequence of integer numbers $\{s_1, s_2..s_p\}$ is a function from $\boldsymbol{I}$ to $I^{'}$ where $\boldsymbol{I}$ and $I^{'}$ are two sets of positive integers.*

Definition 2 ***Subsequence[7].*** *If S=$\{S_n\}$, n=1, 2..∞ is a sequence of integer numbers and N=$\{n_i\}$, i=1, 2..∞ is subsequence of the sequence of positive integers, then the composite function $S \circ N$ is called subsequence of S.*

Definition 3 ***Length of a subsequence.*** *Number of elements of a subsequence is the length of a subsequence.*

Definition 4 ***Block, Block Length.*** *A finite number of binary digits forms a block. Number of bits in a block is called the* ***block length****.*

Definition 5 ***Leader.*** *Leaders are cluster representatives obtained by using Leader Clustering algorithm[4]. The clustering algorithm is explained in Section 3.3.*

Definition 6 ***Support.*** *Support of a feature, in the current work, is actual number of patterns in which the feature is present[6, 3]. Minimum support is referred to as ϵ.*

We use (a) pattern and transaction and (b) item and feature interchangeably in the current work. Following sections describe some of the important steps used.

3.1 Distinct Subsequences

The concepts of sequence, subsequence, length of subsequence are used in the context of demonstrating compactness of pattern. For example, consider a pattern containing binary features, as (0111 0110 1101 .. 0). Consider a block of length 4 and convert each subsequence to decimal code. Now the pattern will contain three blocks each of length of 4-bits, as (7, 6, 13...2). Let each set of three such codes form a subsequence, e.g., (7, 6, 13). In the training set all such distinct subsequences are counted. They form distinct subsequences. Original data of 6000 training patterns consists of 6000*192 features. When arranged as discussed in the current section and distinct subsequences are computed, they turn out to be 690. To elaborate further, consider two most frequent distinct subsequences, (0, 6, 0) and (0, 3, 0) in the entire data across all classes. Subsequence (0, 6, 0) repeats 8642 times and (0, 3, 0) repeats 6447 times. As the minimum support value, ϵ is increased, some of the feature values (binary) would be set to zero. This will lead to reduction in the number of distinct subsequences, as we would show later.

3.2 Computation of Frequent Features

This is intended to examine whether all features help discrimination. Number of occurrences of a feature in training data is computed. If the number is less than given ϵ, value of the features is set to be absent in all the patterns. The remaining features in the training data form 'frequent features'. Value of ϵ depends on the amount of training data, such as class-wise data of 600 patterns each or full data of 6000 patterns.

3.3 Computation of Leaders

The leader clustering algorithm [4] starts with first pattern as the first leader. With the given dissimilarity threshold, the algorithm scans the training data to place patterns within the same cluster. As a pattern that lies beyond the dissimilarity threshold is found, it is selected as next leader. The number of leaders depends on the value of dissimilarity threshold, η.

The leaders are considered as prototypes and they alone are used further, either for classification or for computing frequent items, depending on the adopted approach.

3.4 Classification of Validation Data

The prototypes containing frequent features are used for classifying validation data using k-NNC. Different approaches are followed to generate prototypes. They are described in the following section.

4 Considered Approaches

1. Consider Patterns containing Frequent Items alone
   (a) Identify frequent items for the given support considering (i) class-wise data and (ii) entire training data
   (b) Classify validation data with training data containing frequent features using kNNC
   (c) Compute number of distinct subsequences as an indicator of compactness of the pattern representation
2. Consider Cluster Representatives Alone
   (a) Generate leaders considering (i) class-wise data and (ii) full training data as two approaches
   (b) Classify validation data with leaders using kNNC
3. Computation of Frequent Items followed by Clustering
   (a) Identify frequent items using (i) class-wise data and (ii)the entire data and for the given ϵ
   (b) Generate leaders among the training data containing frequent items
   (c) Classify validation data with leaders containing frequent items using kNNC
4. Clustering followed by Frequent Item Generation
   (a) Generate leaders considering (a) class-wise data and (b) entire training data together
   (b) Identify frequent items in the leader patterns for the given η
   (c) Classify validation data with leaders using kNNC.

5 Discussion of Results

Extensive experimentation is carried out. Summary of results are provided in Table-1.

In Approach-1, we consider all patterns(η=0) and frequent items are identified by considering support values (ϵ) from 0 to 200 for class-wise dataset handling. In case of full dataset ϵ is changed from 0 to 2000. Thus the number of effective items(or features) get reduced per pattern. This in turn results in reduction in distinct subsequences. For the case where best classification accuracy is obtained with validation data, the number of distinct subsequences in case of class-wise data is 507 out of 669 features and 450 out of 669 features. The Classification Accuracies(CA) with test data for class-wise data and full dataset are 92.32% and 92.05% respectively.

In Approach-2, only prototypes are considered(ϵ=0). The distance threshold values are changed from 0.0 to 5.0 in both the cases and prototypes are computed using leader clustering algorithm. For the best case with validation data, the C's with test data are 93.31% and 92.26% respectively. Observe that in this approach, the number of distinct features remain as in the original data.

In Approach-3, frequent items are computed first on the original data and then clustering is resorted to. Frequent items are computed with ϵ values ranging

Table 1. Results with each Approach

Sl No.	Approach	Description	ϵ	η	Proto-types	Distinct features	CA(kNNC) Valdn Data	CA(kNNC) Test Data
1	1	Class-wise data	160	0	6000	507	92.52%	92.32%
2	1	Full data	450	0	6000	361	92.09%	92.05%
3	2	Class-wise data	0	3.1	5064	669	93.14%	93.31%
4	2	Full data	0	3.1	5041	669	93.13%	92.26%
5	3	Class-wise data	40	3.1	5027	542	93.43%	93.52%
6	3	Full data	190	3.1	5059	433	93.58%	93.34%
7	4	Class-wise data	180	3.1	5064	433	93.58%	93.34%
8	4	Full data	300	3.1	5041	367	93.58%	93.52%

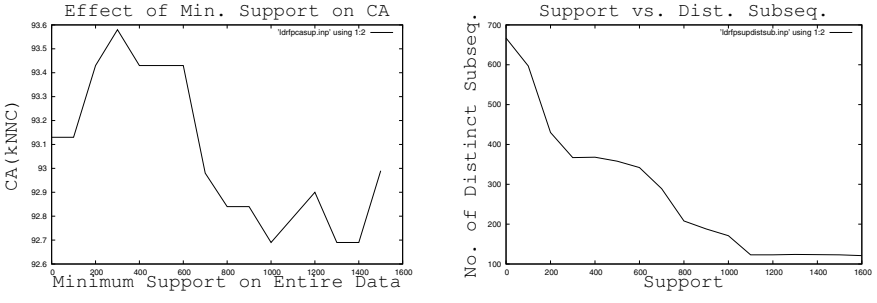

Fig. 2. (a)Support on C.A.(b) Support Vs. Dist. Subsequences in Approach-4 Full Data

from 0 to 200 in steps of 5.0. Subsequently, in each such case, prototypes are computed using leader clustering algorithm. The distance threshold values (η) are changed from 0 to 5.0 in steps of 1.0. The CAs obtained with test data, corresponding to best cases with validation datasets for class-wise data and full dataset are 93.52% and 93.34% respectively. Observe that number of distinct features of these two cases are 542 and 433.

In Approach-4, clustering is carried out first followed by frequent item computation. The CAs with test data corresponding to best cases with validation data are 93.34% and 93.52% for class-wise and full datasets respectivley. Further there is reduction in number of prototypes from original 6000 to 5064 and 5041 for each of these two cases. Significantly, the reduction in distinct subsequences is 433 and 367 respectively.

The number of distinct subsequences can be taken as an indicator of compactness. Another significant result is that there is no **significant reduction in Classification Accuracy(CA) even with good reduction in number of distinct subsequences**. This can be observed from Figure 2, corresponding to Approach-4 with full training data. Fig.2(a) displays CA for various values

of support considering entire data and distance threshold of 3.1. Observe that CA with support of 300 reaches maximum. Fig.2(b) displays reduction in the number of distinct subsequences with increasing support considering full data.

6 Summary

With the objective of handling large data classification problem efficiently, a study is considered to examine the effectiveness of prototype selection and feature selection independently as well as in combination. In the process of the activity, the training data is considered both fully as well as class-wise. kNNC is considered for classifying given large, high-dimensional handwritten data. Extensive exercises are carried out and results are presented.

Important contributions of this work have been to show that combination of prototype and feature selection leads to better Classification Accuracy. Clustering data containing frequent features has provided good amount of compactness for classification. Such compactness did not result in significant reduction in classification accuracy. It is proposed to extend the current work to study the impact of maximum support of an item on compactness of representation as well as for classification.

References

1. Richard O. Duda, Peter E. Hart, David G. Stork.: Pattern Classification John Wiley & Sons, Inc, New York. (2002)
2. Pedro Domingos Occam's Two Razors: The Sharp and the Blunt.: American Association for Artificial Intelligence (1998)
3. Han, J., Pei, J., and Yin, Y. Mining frequent patterns without candidate generation Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data(SIGMOD'00), Dallas, TX, May 2000 (2000) 1–12
4. H.Spath. : Cluster Analysis - Algorithms for Data Reduction and Classification of Objects Ellis Horwood Limited,West Sussex, UK (1980)
5. Ravindra Babu, T., Narasimha Murty, M., Agrawal, V.K. Hybrid Learning Scheme for Data Mining Applications Presented at Conference on Hybrid Intelligent Systems, HIS04, December 06-08, Kitakyushu, Japan (2004)
6. Agrawal, R., Imielinski, T., and Swami, A. : Mining association rules between sets of items in large databases Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data(SIGMOD'93) 207–216, Washington,DC, May 1993 (1993)
7. Richard R. Goldberg : Methods of Real Analysis Oxford & IBH Publishing Co., New Delhi (1978) 24–25
8. Belur V. Dasarathy, J.S.Sanchez : Concurrent Feature and Prototype Selection in the Nearest Neighbour Decision Process Proc. of 4th World Multiconference on Systemics, Cybernetics and Informatics, Vol.VII, Orlando(USA), ISBN 980-07-6693-6 (2000) 628–633.